CMPSCI 250: Introduction to Computation

Lecture #17: Proofs by Mathematical Induction

David Mix Barrington

2 March 2012

Proofs by Mathematical Induction

- Induction as a Proof Rule
- Example: Sum of First k Odd Numbers is k^2
- Common Features of Inductive Proofs
- Example: 2^n Binary Strings of Length n
- Example: 2^n Subsets of an n-Element Set
- Why is Induction Valid?
- Some Counterintuitive Aspects of Induction

Induction as a Proof Rule

- Formally, the Law of Mathematical Induction is just a rule that if we have proved certain statements, we are allowed to claim certain additional statements.
- To use **ordinary induction** (our topic today), we need a predicate P(x) that has one free variable of type `natural`.
- If we prove both "P(0)" and "$\forall$x: P(x) → P(x+1)",
- Then we may conclude "$\forall$x: P(x)".
- Let's look at a simple example.

Example: Sum of First k Odd Numbers is k^2

- The first odd number is $1 = 2 \times 1 - 1$, the second is $3 = 2 \times 2 - 1$, the third $5 = 2 \times 3 - 1$, and in general the k'th odd number is $2k - 1$. (We should actually prove *this* by induction, but there's a technicality because we can't start at 0.)

- We can see that $1 = 1^2$, $1 + 3 = 2^2$, $1 + 3 + 5 = 3^2$, $1 + 3 + 5 + 7 = 4^2$, and so on. We'll let P(k) be the statement "the sum of the first k odd numbers is k^2".

- Proving P(0) is easy -- it says "the sum of the first 0 odd numbers is 0^2", which is true because any empty sum is 0.

- Now we let x be arbitrary and assume that P(x) is true. So the sum of the first x odd numbers is x^2. The sum of the first x+1 odd numbers is the sum of the first x, plus the x+1'st odd number which is $2(x+1) - 1 = 2x + 1$. So (still assuming P(x), we get that the sum of the first x+1 is $x^2 + (2x + 1) = (x+1)^2$.

- Because we proved $P(x) \rightarrow P(x+1)$ for arbitrary x, we are done.

Common Features of Inductive Proofs

- We first proved a **base case** -- the statement P(0) that we get by substituting 0 for x in the statement P(x). Base cases are usually easy to prove.

- We then began the **inductive step**, which is the proof of $P(x) \rightarrow P(x+1)$ for arbitrary x. We assume the truth of P(x), called the **inductive hypothesis**.

- Proving the inductive step usually relies on the fact that P(x) and P(x+1) are related statements. In this case, as with most cases involving sums, P(x+1) talked about a sum that was the same sum that occurred in P(x), plus one more term. So P(x)'s statement about the first sum was useful for us.

- Once we have proved P(x+1) we have completed the inductive case, and then the Law of Mathematical Induction allows us to conclude $\forall x$: P(x).

- Be careful of *types*! "P(x)" is a *boolean*, not a number. If you have a number that is important to P(n), call it S(n) and let P(n) talk about it, but it *isn't* P(n).

Example: 2^n Binary Strings of Length n

- Our next two examples are two similar **counting problems**. In CMPSCI 240 you will learn several general rules for solving counting problems, and these rules can all be proved by mathematical induction.

- We know that there is $1 = 2^0$ binary string of length 0, namely λ. There are $2 = 2^1$ of length 1 ("0" and "1"), and $4 = 2^2$ of length 2 ("00", "01", "10", and "11") We seem to have a general rule that there are 2^n binary strings of length n. To prove this by induction, we let P(n) be the statement "there are exactly 2^n binary strings of length n".

- P(0) is true because there is exactly one empty string. Assume that P(n) is true. Consider all the binary strings of length n+1. Each is either of the form w0 or of the form w1, where w is a string of length n. There are thus *exactly two* strings of length n+1 for each string of length n. The number of strings of length n+1 is thus $2 \times 2^n = 2^{n+1}$. Thus P(n+1) is true (assuming that P(n) is).

- We have completed the inductive step and thus proved ∀x: P(x) by induction.

Example: 2^n Subsets of an n-Element Set

- Let's now prove that any set with n elements has exactly 2^n subsets. We first pick our statement P(n) as "$\forall S: |S| = n \rightarrow S$ has exactly 2^n subsets".

- P(0) says that any set of size 0 has exactly $2^0 = 1$ subset. This is true because a set is a subset of the empty set if and only if it is empty, and there is exactly one empty set.

- Now assume that P(n) is true. To prove "$\forall S: |S| = n+1 \rightarrow S$ has 2^{n+1} subsets", we let S be an arbitrary set of size n+1.

- The key step is to find a set of size n. Let x be any element of S and let $T = S \setminus \{x\}$. Then P(n) tells us that T has exactly 2^n subsets. We can classify the subsets of S into two groups. All subsets of T are also subsets of S. Also if R is any subset of T, $R \cup \{x\}$ is also a subset of S. We have exactly two subsets of S for each subset of T, so there are exactly $2 \times 2^n = 2^{n+1}$ subsets of S.

A Digression: Combinatorial Proofs

- These last two proofs are remarkably similar. Not only is the number of binary strings of length n the same as the number of subsets of an n-element set, the two numbers seem to be 2^n for the *same reason*.

- **Combinatorics** is the study of **counting problems**, determining the size of finite sets (usually parametrized families of finite sets). The holy grail of combinatorics is the **combinatorial proof** -- a demonstration that there is a **bijection** from one set to another and thus that the two sets have the same size.

- In this case we could label the elements of our n-element set as {0, 1, ..., n-1} and map any subset X to the binary string w of length n, such that `w.charAt(i)` is equal to 1 if $i \in X$ and to 0 otherwise. This map has an inverse (where f(w) is the set of indices of w that have a 1) and is a bijection.

- You'll see much more of this sort of thing in CMPSCI 240 and CMPSCI 575.

Why is Induction Valid?

- Formally, we have adopted the Law of Mathematical Induction as part of our definition of the naturals, so if you don't accept it, you are talking about some potentially different number system.
- We can use metaphors to help understand induction -- if we have a set of dominoes arranged so that domino i will always knock over domino i+1, and we push over domino 0, all of them will be knocked over.
- You can think of an induction proof as instructions to construct an ordinary proof. If I want to prove P(4), for example, I have P(0) from the base case, and $P(0) \rightarrow P(1)$, $P(1) \rightarrow P(2)$, $P(2) \rightarrow P(3)$, and $P(3) \rightarrow P(4)$ by Specification on the inductive step. I could prove P(4) directly by using Modus Ponens four times. For that matter I could prove any P(n) directly by using Modus Ponens n times, if I have a valid induction proof.

Some Counterintuitive Aspects of Induction

- An induction proof may appear to use circular reasoning, because in the middle of trying to "prove P(n)", you "assume that P(n) is true". But if you look carefully at the scopes, you see that you are assuming P(n) in order to prove P(n) → P(n+1), in the usual way for a direct proof -- something very different from proving P(n) *without conditions*.

- It's a bit strange to "reduce" the problem of proving ∀x: P(x) to the problem of proving ∀x: P(x) → P(x+1), which is a more complicated statement of the same type. But the latter is usually easier to prove because P(x) is of use in proving P(x+1), while in the former you would have to prove P(x) without conditions.

- Adding conditions to a statement can make it *easier* to prove. If you need some condition Q(n) in order to prove P(n+1), you can use it as long as you can both prove Q(0) in the base case and prove Q(n+1) in your inductive case. Your new induction proves ∀x: P(x) ∧ Q(x) by ordinary induction.